VŠB TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Implementace řídicí jednotky automatického kotle na platformě Raspberry Pi

Implementation of automatic boiler control unit on Raspberry Pi platform

Marián Šimo

Bakalářská práce

Vedoucí práce: Ing. Daniel Sříbný

Ostrava, 2021

Abstrakt

Táto bakalárska práca sa zaoberá tvorbou aplikácie riadiacej jednotky pre automatické kotle. Práca je rozdelená na teoretickú a praktickú časť. Teoretická časť sa venuje analýze a požiadavkom potrebných na oboznámenie problematiky, a praktická časť sa venuje implementácií a testovaniu riadiacej jednotky. Výsledný produkt je pripravená open source aplikácia schopná ovládať kotol na jeden typ paliva

Kľúčové slová

raspberry pi, automatický kotol, riadiaca jednotka, python, gpio

Abstract

This bachelor thesis deals with the creation of a control unit application for automatic boilers. The work is divided into theoretical and practical part. The theoretical part deals with the analysis and requirements needed to acquaint the problem, and and the practical part deals with the implementation and testing. The resulting product is an open source application capable of controlling a boiler for one type of fuel.

Keywords

raspberry pi, automatic boiler, control unit, python, gpio

Poďakovanie

Na tomto mieste by som sa chcel poďakovať vedúcemu práce, pánovi Ing. Danielovi Stříbnemu za odbornú pomoc a optimistický prístup pri tvorbe tejto práce

Obsah

Zoznam použitých skratiek a symbolov

HTML	–	Hyper Text Markup Language
TUV	–	Teplá užitková voda
API	–	Application programming interface
GPIO	–	General-Purpose Input/Output
IoT	–	Internet of Things
MQTT	–	Message Queuing Telemetry Transport
PWM	–	Pulse-Width Modulation
SPI	–	Serial Peripheral Interface
CSS	–	Cascading Style Sheets
AJAX	–	Asynchronous JavaScript And XML
USB	–	Universal Serial Bus
ADC	–	Analog-to-digital converter
IP	–	Internet Protocol address
V	–	Jednotka napätia
C	–	Stupeň celsia

Zoznam obrázkov

Zoznam tabuliek

Zoznam výpisov zdrojového kódu

Kapitola 1

Úvod

Cieľom tejto práce je poskytnúť zhrnutie a prehľad spôsobov vykurovania pomocou automatických kotlov a ich ovládanie cez riadiacu jednotku. Práca je rozdelená do niekoľkých častí.

V prvej kapitole sa rieši problematika kotlov. Nachádza sa tu oboznámenie s rôznymi typmi kotlov na rôzne palivá. Vysvetľujem základný princíp regulátoru pre automatické kotle. Pre každý typ paliva je venovaná malá podkapitola na priblíženie použitia, a základného ovládania jednotlivých kotlov. Taktiež sa opieram o existujúce riešenie na ukázanie industriálneho štandardu.

V nasledujúcej kapitole sa venujem požiadavkom na splnenie cieľa tejto práce a hlavne analýze systému. Je tu zobrazený vlastný model, podľa ktorého bola práca implementovaná. Podľa tohoto modelu boli vybraté komponenty na ovládanie. Bližšie popisujem platformu Raspberry pi ktorá je základným pilierom tohoto systému. V tejto kapitole sú taktiež schémy zapojenia vybratých súčiastok a vývojovej dosky Raspberry pi.

Tretia kapitola sa venuje implementácií a ukážke zdrojových kódov z aplikácie. Kapitola obsahuje zoznam použitých technológií a zdôvodnenie ich výberu. Taktiež obsahuje vysvetlenie štruktúry výsledného programu. Na konci kapitoly je ukážka grafického dizajnu webových stránok so stručným popisom.

V poslednej kapitole sa venujem testovaniu riadiacej jednotky. Obsahuje obrázky ktoré zachytávajú chod základnej časti programu.

Kapitola 2

Analýza problému

2.1 Teória problematiky kotlov

Kotol je kovová konštrukcia ktorej úloha je zohrievať vodu alebo inú tekutinu. Táto zohriata tekutina následne opúšťa kotol v kvapalnom alebo parnom stave pre použitie v rôznych aplikačných sférach, napríklad pri tvorbe energie alebo ako zdroj tepla pre vykurovanie. Takéto kotle používajú ako zroj rôzne typy palív. Delíme ich na sa na pevné, kvapalné a plynné a podľa tohoto delenia každý typ kotla používa rôzne komponenty na správne fungovanie. Najrozšírenejšie sú práve plynné a pevné, lebo sa bežne využívajú v domácnostiach na vykurovanie rodinných domov. Zvolenie správneho kotla závisí od veľkosti zariadenia a požadovaného výkonu v ktorom bude nasadený. Napríklad na vykurovanie malej chatky by plne postačil kotol s nízkou výkonnosťou.

Automatické kotle sú charakteristické práve tým, že pomocou riadiacej jednotky sú schopné ovládať neprerušované dodávanie paliva a tým uľahčiť užívateľovi používanie kotla. Užívateľ sa nestará o vačšinu procesov a len zabezpečuje naplennie zásobníka paliva, nastavuje požadovaný výkon, robí údržbu a zasahuje len v prípade vzniknutého chybového stavu. Riadiaca jenotka ovláda posuvník (či už šnekový alebo valcovitý) a vďaka plynulému dodávaniu paliva majú automatické kotle optimálne podmienky horenia, nastavenie stability chceného výkonu, vysokú účinnosť spaľovania a nízku produkciu škodlivých emisií.

Riadiaca jednotka alebo regulátor je multifunkčné zariadenie ktoré ma viacej kľúčových úloh. Automaticky udržuje žiadanú teplotu kotla vďaka tomu že vlastne riadi spaľovací proces. Reguluje podavač paliva a ventilátor so správnym časovancím procesom. Ovláda obehové čerpadlá ktoré čerpajú vodu do radiátorov, a akumulačnej nádrže kde sa udržiava žiadaná teplota teplej užitkovej vody. V riadiacej jednotke sú taktež zabudované rôzne režimy ovládania kotla.

Kotle musia spĺňať technické predpisy a normy ohľadom svojho umiestnenia. Správne umiestnenie kotla závisí hlavne od typu paliva. Od toho sa odvíjajú potrebné zložky pre daný kotol. Napríklad ak sa jedná o kombinovaný kotol (to znamená že zabezpečuje vykurovanie a aj ohrev TUV), je nutné ho umiestniť blízko akumulačnej nádrže aby sa v potrubí zbytočne nestrácala tep-

lota vody. Vykurovacie kotle sa v rodinných domoch ukladajú bežne do pivníc, práve kôli suchosti a praktickému využitiu miestnosti. Výhodou automatického kotla s riadiacou jednotkou je komfort a pohodlie uživateľa, nakoľko nieje nútený zakaždým zísť do kotolne pri potrebnom nastavovaní, ale využie webové rozhranie cez počítač či smartfón.

2.2 Funkčné časti automatického kotla

Ako bolo už vyššie spomenuté automatické kotle sa delia podľa použitého paliva. Nasledujúca časť sa venuje analýze, a základným konštrukčným prvkom jednotlivých kotlov na jasnejšie priblíženie požiadavkov na riadiacu jednotku automatického kotla.

2.2.1 Kotol na kvapalné palivo

Použitie kotla na kvapalné palivo nie je zrovna najbežneším spôsobom vykurovania rodinnych domov. Dôvodom nieje neefektívne spaľovanie, práve naopak účinnosť väčšiny modelov dosahuje 95 percent. Problém nastáva pri nákladoch na palivo. Pri intenzívnom používaní zariadenia môže ročná spotreba paliva dosiahnuť niekoľko ton. Taktiež by bolo nákladné skladovať toto palivo pretože na skladovania paliva sa stavia samostatná budova. Základnou úlohou riadiacej jednotky je udržovanie chcenej teploty. Riadiaca jednotka pre kvapalné kotle obsluhuje obehové čerpadlo ktoré slúži ako prívod paliva zo zásobníkovej nádrže do horáka. Využíva sa vykurovací olej alebo poprípade motorová nafta a preto musí byť palivo bezpečne skladované lebo sa jedná o vysoké horľaviny. Pod vplyvom ventilátora sa zmes rozprašuje a palivová hmla sa v spaľovacej komore vznieti čo generuje teplo [1].

2.2.2 Kotol na plynné palivo

Plynový kotol využíva zemný plyn, ktorý do neho prúdi z plynovodu. Po spustení regulácie kotla sa otvorí ventil a vpustí sa plyn do spaľovacej komory ktorá sa nachádza v kotli. Všeobecne pri spaľovaní plynných substancií (propán-bután, zemný plyn) nevznikajú sadze ani iné pevné častice ktoré by mohli spôsobovať znečistenie životného prostredia. Práve aj kôli tomu sa považuje vykurovanie pomocou plynového kotla za jeden z ekologickejších spôsobov a nachádza sa v mnohých domacnostiach. Čo sa týka riadiacej jednotky [2].

2.2.3 Kotol na tuhé palivo

Kúrenie pomocou dreva bolo od nepamäti najrozšírenejším spôsobom získavania tepla. V dnešnej dobe sú ako palivo rozšírené drevenné pelety, brikety a uhlie. Kotol na tuhé palivo má blízko seba zabudovaný zásobník s vysokým objemom v ktorom sa vybrané palivo uchováva. Toto palivo sa presúva pomocou posuvníka kde v uzavretom priestore s pridaním vzduchu dochádza k spaľovaniu. Riadený spaľovací proces je monitorovaný riadiacou jednotkou a zabezpečuje optimálnu produkciu

emisií. Po následnom nahriatí na žiadanú teplotu sa automaticky spustí obehové čeradlo ktoré púšťa TUV do radiátorov. Kotol v stave bez elektrickej energie dokáže dokonca kúriť na prídavnom rošte, teda používateľ si môže ručne prikladať poprípade spaľovať rôzne spáliteľné odpady z domácnosti [3].

2.3 Existujúce riešenie

Na trhu sa nachádza množstvo rôznych kotlov a riadiacich jednotiek k nim. Väčšinou býva tak, že rôzne značky kotlov sú budované na kopatibilitu s danou riadiacou jednotkou. Ako príklad sa dá uviesť kotol Benekov b20 ktorého riadiaca jednotka je EM 800R4. Táto riadiaca jednotka je postavaná na báze vývojovej dosky Arduina. Aj keď platformy ako Arduino a Raspberry pi nie sú určené k tomu aby pracovali niekoľko rokov vkuse, využitie v tejto sfére sa hlavne odráža na pomere cena/výkon, nakoľko senzory dostupné na trhu sú väčšinou pomerne lacné. Táto riadiaca jednotka pochádza od Poľskej firmy Plum.

Ako ďalší príklad sa dá uviesť jednotka Siemens Climatix čo je priemyselný štandard od firmy Siemens. V porovnaní s EM 800R4 je niekoľkonásobne drahší a má akúsi fázy logiku, to znamená dokáže predvídať určité situácie a dokáže prispôsobiť kotol čo najefektívnejšie. Avšak čo sa týka webového rozhrania, tak v porovnaní s Plumom má len plané html ktoré zobrazuje namerané hodnoty.

Každá riadiaca jednotka musí uživateľovi príjemným spôsobom umožniť ovládanie kotla. Cieľ práce je naprogramovať alternatívnu open source jednotku k týmto komerčným riešeniam, aby uživateľ v prípade potreby mohol upraviť kód podľa potreby naladenia vlastného kotla.

Kapitola 3

Vlastný návrh riešenia

3.1 Analýza požiadavkov informačného systému

Táto práca sa zameriava na návrh a implementáciu riadiacej jednotky pre automatické kotle na platforme Raspberry Pi. Výsledné riešenie má byť modulárneho typu. To znamená že systém bude poskytovať možnosť rozšírenia v podobe modulov, ktoré budú vybrané podľa typu kotla o ktoŕy sa jedná. Software tejto jednotky bude ovládať potrebné vstupy a výstupy. Konkrétne ovládanie kotla bude ladené podľa daných vstupov, výstupov a závislostiach medzi nimi, čiže podľa konkrétneho prispôsobenia kotla. Výsledný produkt by mal obsahovať webové uživateľské rozhranie na sledovanie a konfigurovanie stavu systému.

- Výber správneho hardware na ovládanie a poprípade monitoring všetkých potrebných častí ktoré riadiaca jednotka má ovládať. Z toho vyplýva že sa bude jednať o programovanie rôznych senzorov.Voľba hardwaru však musí byť zvolená na základe prostredia v ktorom bude tento hardware nasadený. Všetky tieto vstupy a výstupy treba ale ovládať s niečím, s čím je Raspberry Pi kompatibilné.

- Implementácia vlastného open source systému jednotky na jeden zvolený konkrétny typ paliva

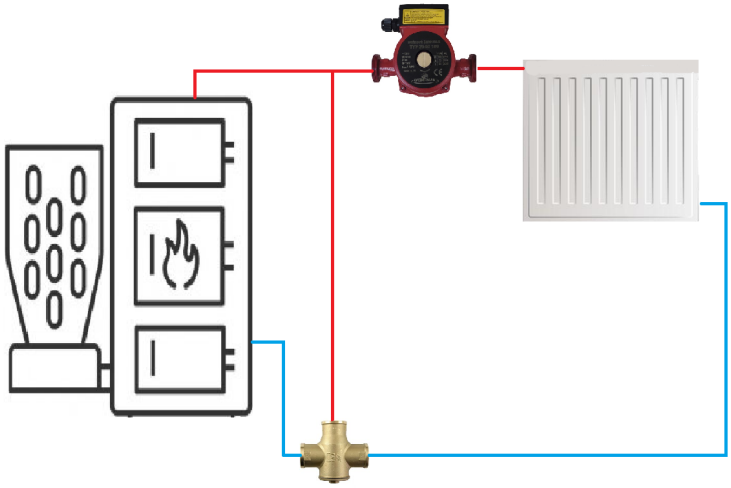
- Implementácia API pre daný systém. To znamená vytvorenie knižnice ktorá bude obsahovať všetky použité programátorské prvky pre prácu s týmto systémom ktoré budú otvorené pre širokú verejnosť.

- Vytvorenie webových stránok pre monitorovanie stavu riadiacej jednotky. Uživateľ by mal mať možnosť remotne konfigurovať nastavenia riadiacej rednotky

- Odtestovanie tejto jednotky v praxi, a tak aj celkovú funkčnosť hardwaru so softwarom

3.2 Vlastný model

Jeden z cieľov tejto práce je vybrať si jeden typ kotla podľa paliva a naprogramovať riadiacu jednotku ktorá je schopná ovládať tento kotol. Zvolil som si automatický kotol na tuhé palivo s posuvníkom paliva. Regulátor bude mať však možnosť rozšírenia implementácie pre ostatné spomenuté typy palív.



Obr. 3.1: Jednoduchý model riešenia

Riešenie je závislé od typu hydraulického zapojenia kotla. Na obrázku je znázornený model použitý v tejto práci. Jedná sa o zapojenie pomocou trojcestného ventilu. V komerčných riešeniach je možností zapojení viac. Okrem zapojenia podľa modelu je taktiež veľmi populárne zapojenie pomocou štvorcestného ventilu ktorý má dva topné okruhy. Pre každý okruh je jedno čerpadlo ktoré ovláda cirkuláciu vody buď pre dom alebo kotol, a reguluje sa pomocou prepustnosti ventilu.

Podľa použitého modelu je možné spozorovať že topný okruh má čerpadlo jedno, a to pre kotol aj pre dom. Riadiaca jednotka bude teda obsluhovať toto obehové čerpadlo ktoré pumpuje teplú vodu do radiátorov. Aby regulácia prebiehala v poriadku, je treba monitorovať teplotu vody odchádzajúcej z kotla, a teplotu spätného toku. Samozrejme riadiaca jednotka má za úlohu kontrolovať spaľovací proces v komore kotla čo najefektívnejším možným spôsobom.

3.3 Výber komponentov

3.3.1 Raspberry Pi

Raspberry pi je malý jednodoskový počítač s rozmermi kreditnej karty ktorý beži na báze operačného systému Linux. Raspberry Pi má všetky potrebné vstupy na zapojenie periférnych zariadení. S použitím USB klávesnice, myši a HDMI monitora je možné bezproblémov použiť Raspberry Pi ako plnohodnotný počítač. Taktiež obsahuje 3,5mm zvukový konektor ako výstup reproduktora a má ethernetový port na pripojenie internetu. Problém nastáva v úložisku ktoré má veľkosť RAM od 2GB - 8GB (podľa vybratého modelu). Z toho dôvodu je nutné použiť vstup na micro SD kartu s minimálne 2GB kapacitou na ktorej je nainštalovaný operačný sytém. Raspberry pi je veľmi cenovo dostupná vývojová platforma, kde sa ceny pohybujú od 45€ - záleží od zvoleného modelu a zvolenej veľkosti pamäte [4].

Raspberry pi 4 je novšou generáciou tohoto jednodoskového počítača. Ponúka novinky v podobe možnosti pripojenia dvoch 4k monitorov, väčšej rýchlosti procesora a pamäte, ale hlavne je kompatibilný so svojimi predošlími verziami ako napríklad Raspberry Pi 3 alebo Raspberry Pi 3 B+.

V projekte je využívaný ako logická jednotka ktorá ma na startosti ovládanie senzorov pomocou GPIO pinov. GPIO na Raspberry Pi nám umožňuje zapojiť až 26 senzorov čo je na cieľ tejto práce postačujúce. Na rozdiel od mikrokontroléru máme možnosť využiť Raspberry Pi ako klient-server komunikáciu.

Tabuľka 3.1: Parametre použitej dosky Raspberry Pi 4

Rozmery	85,60 mm × 56,5 mm
Procesor	Broadcom BCM2711 1.5 GHz quad-core ARM Cortex-A72 ARM v8 64-bit SoC
Pamäť	typ LPDDR4 veľkosť 2 GB
Interná pamäť	port pre MicroSD
Konektivita	2x USB 2.0, 2x USB 3.2, LAN, 3.5mm jack, micro HDMI
GPIO	40 pinov (spätne kompatibilné s predošlými modelmi)
Zdroj	5v MicroUSB

3.3.2 Operačný systém

Linux je voľne dostupný open source operačný systém ktorý patrí do Unix systémov. Linux znamená samotné jadro ktoré rieši komunikáciu medzi uživateľom a hardwarom - označuje sa ale na označenie celej distribúcie Linuxu.

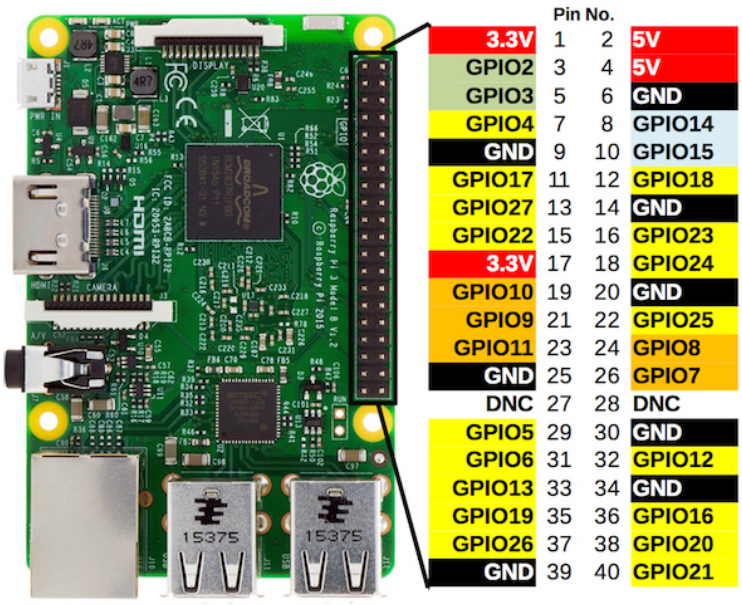
V projekte je využitý základný 32bitový operačný systém ktorý ponúka platforma Raspberry Pi pri inštalácii z oficiálnej stránky 'raspberrypi.org'. Jedná sa o Raspbian, čo je založený na operačnom systéme Debian (Na báze linuxu). Obsahuje predinštalované software ako Python, Sonic Pi, Java,

Mathematica a mnoho ďalších. Aj keď raspbian má rozhranie príkazovho riadku (terminál), tak prichádza s grafickým uživateľským rozhraním. Raspberry Pi nie je limitované na použitie Raspbianu ale má možnosť nainštalovania iných podporovaných operačných systémov ako napríklad Ubuntu mate alebo ArchLinux. Na použitie do projektu mi prišiel Raspbian ako vhodná dostačujúca voľba.

3.3.3 GPIO

Raspberry Pi má na svojom boku 2 rady GPIO pinov po 20. GPIO značí pre general purpose input-output piny, takzvané univerzálne vstupno-výstupné porty. Tieto konektory nám umožňujú pripojiť extérny hardware s ktorým Raspberry Pi komunikuje.

Porty označené ako GPIO majú možnosť byť definované buď ako vstup a výstup a ich stavy môžu byť '1' alebo '0'. Ak sa jedná o vstup tak sa dostane do stavu '1' ak presiahne voltáž 1.7V. Pri výstupe dosiahne stavu logickej jednotky ak prekročí napätie 3,3V.



Obr. 3.2: GPIO porty na Rasperry Pi
[**https://www.bigmessowires.com/2018/05/26/raspberry-pi-gpio-programming-in-c/**]

GPIO sú štandartné piny ktoré sa používajú na komunikáciu vstupno-výstupných dat s Rasperry Pi. Všeobecne nemajú definovaný účel. Zdrojové piny majú výstup 3,3V a 5V DC z Raspberry Pi. GND (ground) sa používa na zapojenie zaradenia do zeme - uzemnenie. DNC (do not connect) väčšinou sa tieto piny nezapájajú [5].

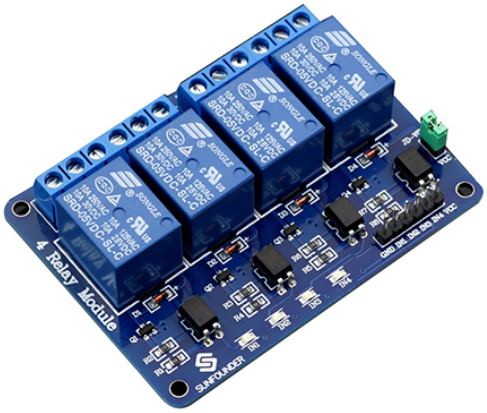
Raspberry pi nám dokopy umožňuje pomocou GPIO pinov zapojiť a ovládať až 26 extérnych zariadení - senzorov [6]. Medzi 40 konektormi sa nachádzajú piny práve na špeciálne použitie ktoré sú popísané v nasledujúcej tabuľke.

Tabuľka 3.2: Ostatné GPIO porty na Raspberry Pi

Špeciálne GPIO piny		
Pin	**Označenie**	**Význam jenotliych pinov**
GPIO 2	SDA	Data line
GPIO 3	SCL	Clock
GPIO 9	MISO	SPI
GPIO 10	MOSI	SPI
GPIO 11	SCLK	SPI
GPIO 14	Tx	Transmit pin for the serial port
GPIO 15	Rx	Recieve pin for the serial port
ID_SC		Reserved for Pi plates
ID_SD		Reserved for Pi plates

3.3.4 Komponenty na ovládanie a monitoring

Pre návrh a implementáciu riadiacej jednotky je nevyhnutný správny výber modulov pre ovládanie a monitoring jednotlivých častí kotla.



Obr. 3.3: 4 Kanálové relé 5V
[**https://aptofun.de/4_Channel_5V_Relay_Module**]

Relé jé elektronická súčiastka ktorá slúži na ovládanie modulov ktoré majú zvyčajne vyššiu voltáž. V mojom projekte slúži na ovládanie jednotlivých obehových čerpadiel a posuvníka paliva. Jednotlivý kanál reprezentuje zapojenie každého komponentu. Spínacie relé funguje vlastne ako spínač - udáva output hodnotu HIGH alebo LOW podľa toho či modulom prechádza prúd alebo nie. Tieto relé moduly sú dostupné v rôznych prevedeniach. Dajú sa zohnať s rôznym počtom kanálov a s rôznou voltážou napríklad 3.3V, 5V, 6V, 12V, 24V. Keďže Raspberry Pi toleruje maximum 5V (GPIO dokonca len 3,3V) je nutné vybrať relé so vstupným signálom do 5V. V opačnom prípade môže dojsť k trvalému poškodeniu vývojovej dosky [7].

Ako parameter do funkcie vstupuje gpio pin ktorý ma byť ovládaný. Konkrétne sa môže jednať o čerpadlo alebo podavač paliva.

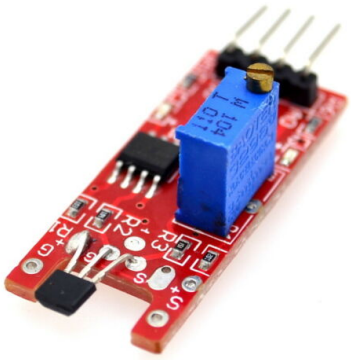
```
import RPi.GPIO as GPIO
import time

def RelayOn(Pin_GPIO):
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(Pin_GPIO,GPIO.OUT)
    try:
        GPIO.output(Pin_GPIO,GPIO.HIGH)
        time.sleep(1)
    finally:
        GPIO.cleanup()

def RelayOn(Pin_GPIO):
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(Pin_GPIO,GPIO.OUT)
    try:
        GPIO.output(Pin_GPIO,GPIO.LOW)
        time.sleep(1)
    finally:
        GPIO.cleanup()
```

Výpis 3.1: Ukážka zdrojového kódu pre ovládanie relé boardu

Obr. 3.4: Tepelné čidlo dallas DS18B20
[**https://www.itead.cc/waterproof-ds18b20-temperature-sensor.html**]

Dallas DS18B20 je tepelný senzor ktorý umožňuje merať teplotu v rozsahu -55 až 125 stupňov Celzia s presnosťou 0,5 stupňa. Použité teplomery budú snímat teplotu vody, podľa ktorej sa budú zapínať jednotlivé obehové čeradlá. Kedže ide o meranie teplej užitkovej vody, najvhodnejšie je použiť vodotesnú verziu tohoto teplomera, ktorý je aj jednoduhší na praktické použitie do danného prostredia.

Tento diditálny teplomer poskytuje merania 9b-12b v stupňoch Celsia. DS18B20 komunikuje cez zbernicu 1-Wire pričom vyžaduje len jeden datový vodič pre komunikáciu s mikroprocesorom, ktorý vlastne nahrádza externý zdroj napájania. Každý teplomer DS18B20 má unikátne 64bit číslo, ktoré umožňuje na jednej zbernici pracovať viacerými teplomermi naraz. S použitím s Raspberry Pi je v celku jednoduchý. Datový vodič sa napojí na jeden vstupno-výstupný GPIO pin a keďže DS18B20 potrebuje napätie 3-5V, je možno použiť buď 3,3V alebo 5V GPIO pin. Potrebný je ale zapojiť pullup rezistor ktorý zabezpečí správne napätie pre fungovanie teplomera. Tento rezistor sa zapája sa medzi datový a napájací vodič. Impedancia tohoto rezistora by mala byť priemerne 5K ohma [8].



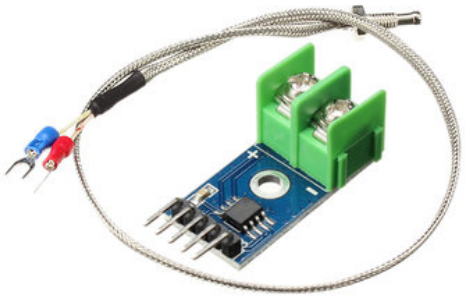
Obr. 3.5: Hallov senzor
[**https://www.ebay.com/itm/1pcs-KY-024-Linear-Hall-Magnetic-Module**]

Hallov senzor je integrovaný obvod, ktorý je schopný detekovať prítomnosť magnetického poľa pomocou hallovho efektu. V projekte ho využívam ako odozvu pre posúvnik šnekového podavača (podavač vieme ovládať pomocou spínacieho relé, ale musíme ho aj monitorovať aby sme vedeli či pracuje ako má alebo aby sme mohli zasiahnuť v prípade chybového stavu). S týmto senzorom sme schopný počítať otáčky, čo nám teoreticky umožňuje vypočítať približný počet spáleného paliva na základe počtu otočení posúvníka. Na rozdiel od platformy Arduino, Raspberry pi neposkytuje ADC, a keďže senzor KY-024 má analogový a digitálny výstup, tak je platforma Raspberry Pi práve limitovaná na použitie toho digitálneho ktorý poukzaje len na prítomnosť magnetického poľa [9].

Ako parameter funkcie je gpio datový pin. Funkcia vracia boolean na základe zisteného magnetického poľa.

```
import RPi.GPIO as GPIO

def IsMagneticFieldDetected(Digital_PIN)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(Digital_PIN, GPIO.IN, pull_up_down = GPIO.PUD_OFF)
    if GPIO.input(Digital_PIN):
        return True
    else:
        return False
```

Výpis 3.2: Ukážka zdrojového kódu pre ovládanie hallovho senzoru



Obr. 3.6: Termočlánok typ 'K' s prevodníkom MAX6675
[**https://www.banggood.com/MAX6675-Sensor-Module**]

Riadiaca jednotka musí vedieť aká teplota sa nachádza v kotli aby bolo jasné či horák horí alebo nehorí. Na získanie tejto teploty nám obyčajné tepelné čidlá nepomôžu, nakoľko tam teplota presahuje bežných 125 stupňov celsia pri ktorej by sa sondy roztavili a preto je nutné použiť určitý druh termočlánku. Termočlánok typu "K"má rozshah merania teplôt od -200 až po 1024 stupňov čo je na meranie teploty spalin viac než dostačujúce. Využíva vznik napätia, ktorý nastáva pri teplotných rozdieloch medzi dvoma rozdielnymi kovmi alebo polovodičmi. To spôsobuje nepretržité prúdenie elektrónov a vzniknuté napätie je rádovo niekoľko mikrovoltov na stupeň Celzia. Keďže termočlánky vytvárajú napätie niekoľko desiatok mV je potrebné použiť zosilnenie.

MAX6675 je prepracovaný prevodník termočlánku na digitálny prevodník, so zabudovaným 12-bitovým analógovo-digitálnym prevodníkom (ADC). Model MAX6675 tiež obsahuje studený spoj kompenzačného snímania a korekcie, digitálny radič a rozhranie kompatibilné so SPI a súvisiacou logikou riadenia. Úloha tohoto prevodníka je digitalizovať namerané volty ktoré majú termočlánky na výstupe pri meraní teplôt, čo nám umožní pracovať s nameranou hodnotou v programe [10].

Funkcia vracia nameranú teplotu z termočlánku

```
import spidev, time

spi = spidev.SpiDev()
spi.open(0, 0)
spi.max_speed_hz = 3900000

def GetTemperatureFromMAX6675()
    t = spi.readbytes(2)
    time.sleep(0.5)
```

```
    msb = format(t[0], '#010b')
    lsb = format(t[1], '#010b')
    r_temp = msb[2:] + lsb[2:]
    t_bytes = "0b" + r_temp[0:13]
    temp = int(t_bytes, base=2)*0.25

    return temp
```

Výpis 3.3: Ukážka zdrojového kódu pre termočlánok s prevodníkom

3.3.5 PWM

Impulzová šírková modulácia je signál ovládaný pomocou prepínaním napätia v určitých časových intervaloch. Je to bežná technika ktorá sa používa na efektívne regulovanie rýchlosti alebo intenzity. Raspberry Pi má túto moduláciu vstavanú vo svojich GPIO pinoch a dajú sa softwarovo ovládať [11].

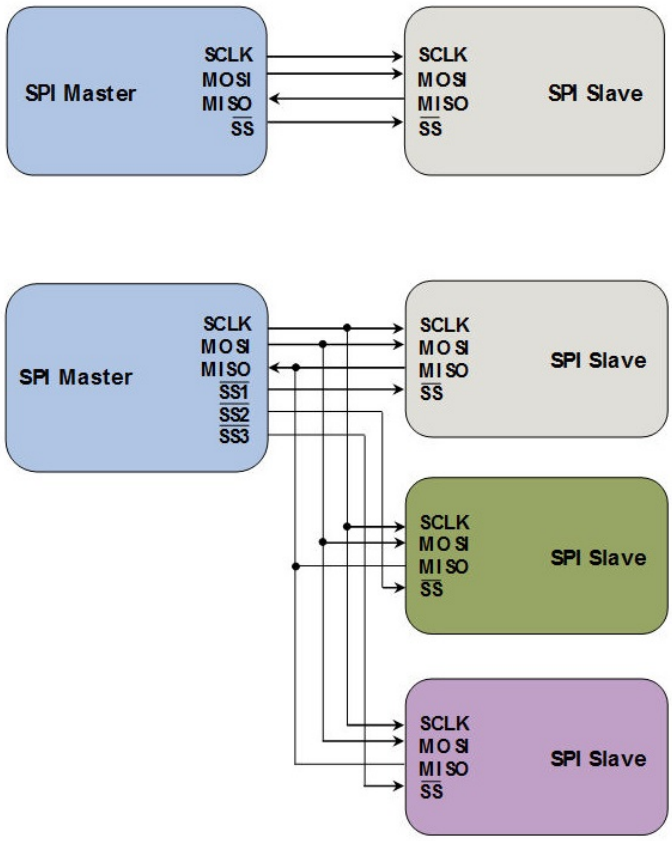
3.3.6 SPI

Sériovo periferné rozhranie je single-master protokol ktorý slúži na komunikáciu medzi kontrolérmi a modulmi. Má štvorvodičovú zbernicu na ktorej komunikuje so zapojenými zariadeniami.

Zobrazené topológie SPI zbernice. Horný obrázok zobrazuje zapojenie jedného zariadenia. Spodný obrázok ukazuje SPI zapojené do viacerých zariadení.

Pre podporu SPI v Pythone je potrebné doinštalovať knižnicu spidev. Na jej inštaláciu bude potrebný balíček python-dev.

```
$ sudo apt-get install python-setuptools python-dev git-core
```



Obr. 3.7: Štvorvodičová SPI zbernica
[**https://www.programmersought.com/article/89994782874/**]

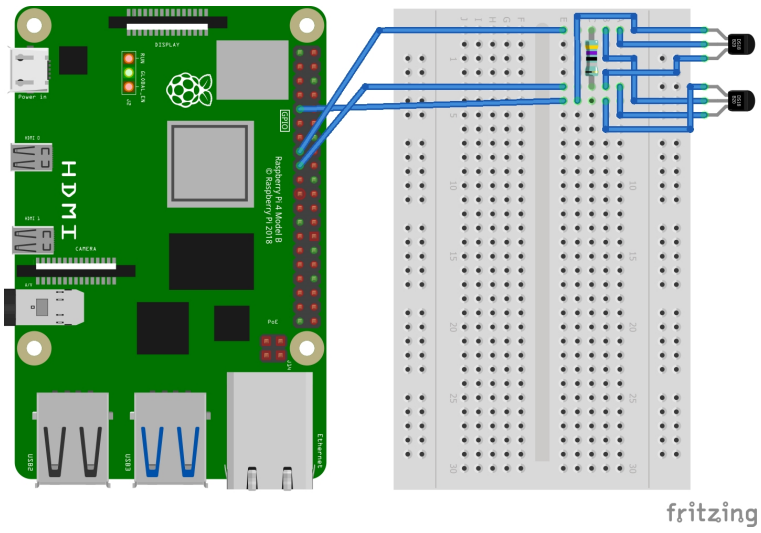
3.3.7 Výber programovacieho jazyka

Raspberry Pi ponúka podporu pre množstvo programovacích jazykov ako napríklad Java, JavaScript, Python, C++. Na túto prácu bol zvolený skriptovací jazyk Python. Python je jeden z najrozšírenejších programovacích jazykov vôbec. Vznikol v roku 1991 a bol vynalezený Guido van Rossuom s cieľom zjednodušiť písanie riadkov kódu a skrátenia ich dĺžky čím je vlastne Python špecifický. Čoraz častejšie sa stáva prvým programovacím jazykom novodobých programátorov. Výhoda tohoto jazyka je možnosť písať objektovo-orientované kódy a aj procedurálne programovanie [12].

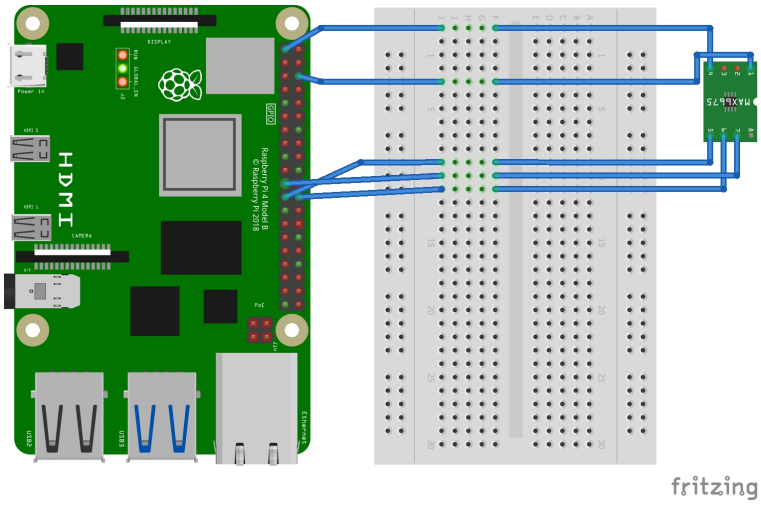
Programovací jazyk Python je predinštalovaný v oficiálnom operačnom systéme Raspbian ktorý ponúka Raspberry Pi. Prichádza v dvoch verziách a to 2.7 a 3.5 čo umožňuje kompatibilitu aj so staršími knižnicami. Rozhodnutie o výbere Pythonu ako zvoleného programovacieho jazyka taktiež podporovalo množstvo dostupných knižníc ktoré v dnešnej dobe internet ponúka na riadenie väčšiny senzorov. Raspberry Pi s pomocou Pythonu je takmer najkompatibilnejší na riadenie a komunikáciu extérnych zariadení.

3.3.8 Schémy zapojenia

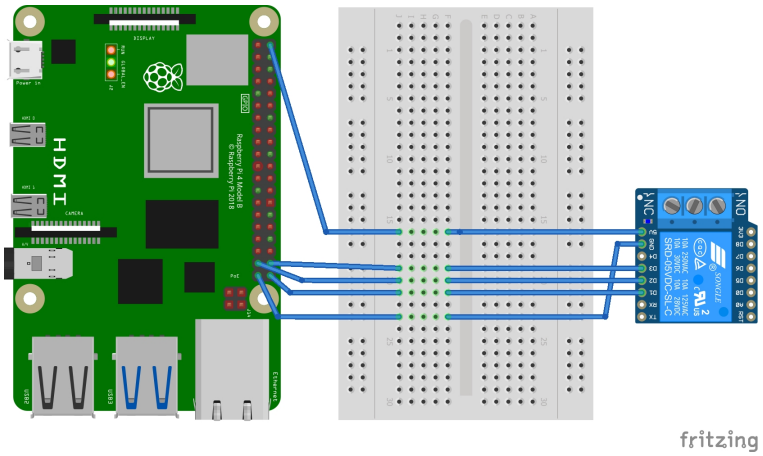
Nasledujúce schémy zobrazujú presné zapojenie senzorov ktoré boli použité v praxi. Kombinácia týchto zapojení tvorí celkový produkt po hardwarovej stránke. (pre vytvorenie schém bol použitý program fritzing)



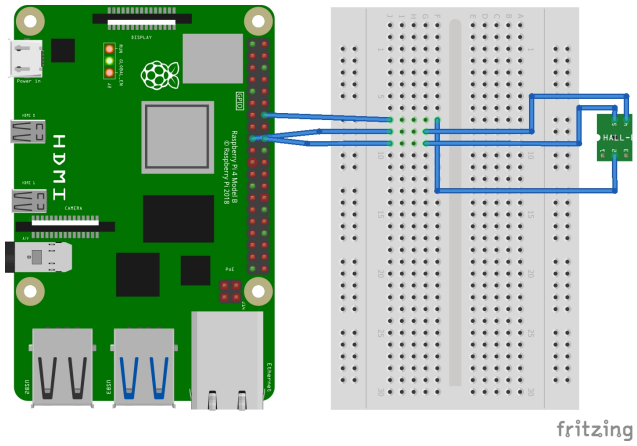
Obr. 3.8: Schéma zapojenia dvoch teplomerov dallas



Obr. 3.9: Schéma zapojenia prevodníku max6675



Obr. 3.10: Schéma zapojenia spínacieho relé a použitia troch kanálov



Obr. 3.11: Schéma zapojenia senzoru hallovho efektu

Kapitola 4

Implementácia systému

4.1 Štruktúra programu

Implementácia je riešená v jednom samostatnom projekte. Využité bolo programovacie prostredie PyCharm, čo je jeno z najpopulárnejších vývojových prostredí pre programovací jazyk Python. Zvolený štýl je funkčné programovanie. Projekt je členený na viacero častí.

- Priečinok *Integration* obsahuje všetky skripty potrebné pre komunikáciu so senzormi pomocou Raspberry pi. Konkrétne sa jedná o ovládanie relé, ventilátora pomocou pwm, termočlánku, hallovho senzoru a teplotných čidiel. Tieto komponenty môžu byť ľubovoľne zamenené a nemajú vplyv na základnú logiku, čo zabezpečuje potrebnú modularitu programu a minimálne zasahovanie do kódu v prípade zmeny súčiastky.

- Priečinok *Devices* obsahuje všetky funkcie pre konkrétne zariadenia potrebné na fungovanie. V tomto prípade sa jedná o čerpadlo, teplomer pre kotol a spätný tok, ventilátor, posuvník paliva, trojcestný ventil, a teplomer na spaliny. Pre prácu využívajú skripty práve z priečinku pre integráciu so senzormi.

- Priečinky *Templates* a *Static* slúžia na uchovávanie všetkých súborov potrebných na prezentáciu webovej stránky. Jedná sa o html súbory a png obrázky.

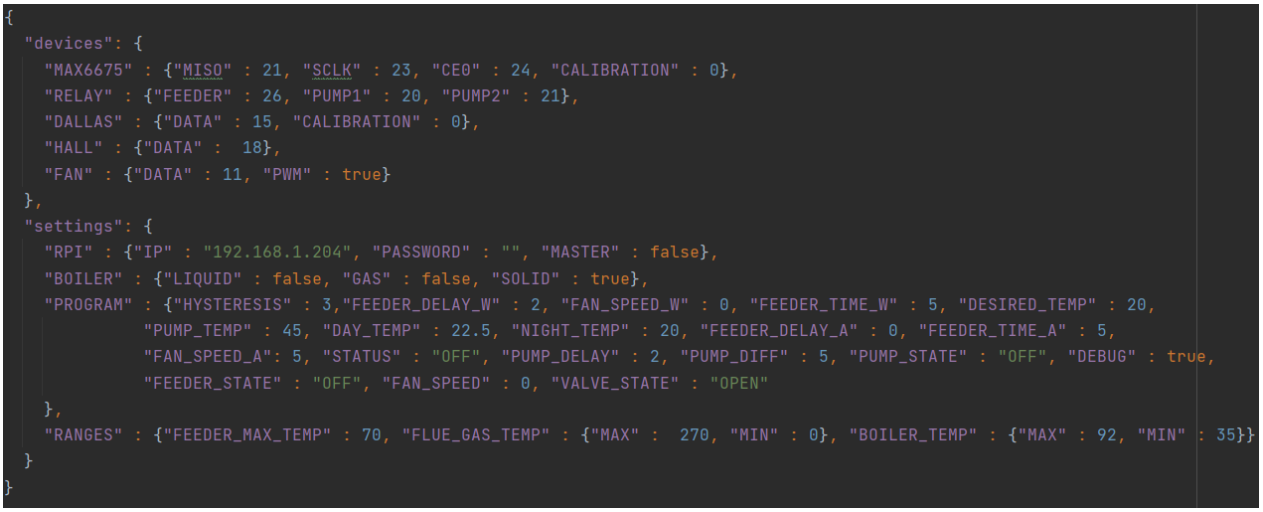
- Priečinok *Diagnostics* obsahuje logovací súbor ktorý zapisuje správanie programu, poprípade zachytáva chybové stavy.

- Súbory *conf.json* a *default_conf.json* sa využívajú ako konfiguračné súbory pre program. Obsahujú všetky potrebné data pre správny chod programu

- Súbor *FileManager.py* obsahuje všetky funkcie potrebné na prácu s konfiguračným súborom.

- Súbor *MqttClient.py* obsahuje funkcie v prípade distribuovaného nasadenia programu.

- Hlavné jadro programu sa nachádza v súbore *Main.py* a spolu so súborom *Status.py* obsahuje všetku logiku spolu pre web a program.



Obr. 4.1: Vzorový konfiguračný súbor

4.2 Použité technológie

4.2.1 Flask

Je framework pre programovací jazyk python ktorý umožňuje vytvorenie webového servera bez žiadnych špeciálnych nástrojov. Server môže bežať nepretržite bez problémov. Nemá žiadnu vrstvu abstrakcie databázy, overovanie formulárov ani žiadne ďalšie komponenty, kde predtým existujúce knižnice tretích strán poskytujú bežné funkcie ako napríklad framework Django. Flask nie je stavaný na vývoj väčších projektov, ale je jednoduhší na použitie. Z toho dôvodu som si ho vybral na tvorbu webového rozhrania pre riadiacu jednotku, nakoľko toto rozhranie bude mať v praxi len pár uživateľov [13].

4.2.2 Css

Je jazyk ktorý slúži na formátovanie dokumentov. Väčšinou sa spája s jazykom html na štýlovanie grafického uživateľského rozhrania. V projekte bol použitý pre dizajn webových stránok.

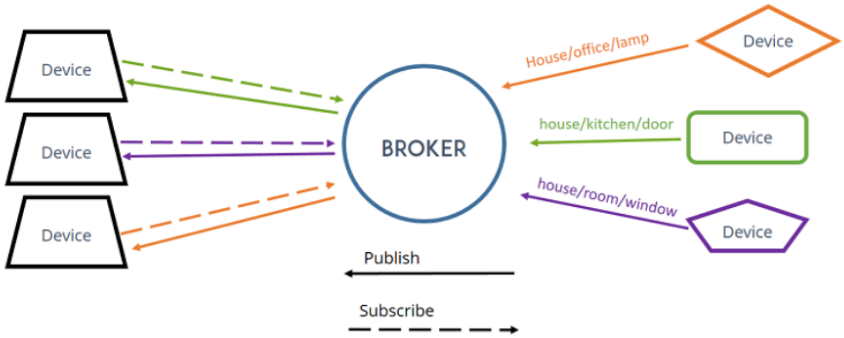
4.2.3 Ajax

Je technológia ktorá umožňuje webovým stránkam asynchrnónnu aktualizáciu výmenu údajov na server a zo servera. To znamená že je možné aktualizovať len časť stránky bez nutnosti opätovaného načítania. Zahŕňa kombináciu integrovaného objektu XMLHttpRequest v prehliadači, JavaScriptu a HTML. V projekte je využívaný na dynamické načítavanie teplotných hodnôt a stavu zariadení na web [14].

4.2.4 MQTT

Je protokol ktorý umožňuje komunikáciu medzi viacerými zariadeniami. Používa návrhový vzor publisher - subscriber. To znamená že existuje jeden centrálny bod (MQTT broker), ktorý sa stará o výmenu správ. Správy sú triedené do tém (topic) a zariadenia buď zdieľajú danú tému (publish), alebo ju odoberajú (subscribe). Broker potom všetky správy s danou témou posiela do zariadení ktoré danú tému odoberali. Prenos funguje pomocou protokolu TPC. Mqtt je bežná voľba a riešenie pre komunikáciu zariadení pre IoT aplikácie [15].

V projekte je využívaný na distribuovanosť systému. Pomocou tohoto protokolu sme schopný zvoliť jeden kotol ako Mastra, a pomocou ním ovládať ďalšie kotle ako Slave. Master zdieľa príkazy s danou témou a Slave kotle tieto témy odoberajú a následne posielaju odozvu pre hlavný kotol.



Obr. 4.2: Princíp mqtt brokera
[**https://randomnerdtutorials.com/cloud-mqtt-mosquitto-broker-access-anywhere-digital-ocean/**]

Funkcia on_message príjma a dekóduje správy zaslané od mastra. správa prichádza vo forme slovníka a podľa kľúča "name"zistí svoju funkciu. Slave následne posiela naspäť odozvu pre svojho mastra.

```
# prijatie spravy od mastra pre slave
def on_message(client, userdata, message):
    msg = str(message.payload.decode("utf-8"))
    try:
        res = ast.literal_eval(msg)
        if res['name'] == 'updateFormData':
            data = res["value"]
            fm.updateFormData(*data)
            publish_msg("Logging", ' updating form data on slave')
    except:
        publish_msg("Loggins", 'Unable to read recieved message')
#inicializacia mqtt klienta pre slave
mqttBroker = "mqtt.eclipseprojects.io" #free online broker
client = mqtt.Client("Slave")
client.connect(mqttBroker)
client.subscribe("Control") #odoberanie sprav s temou Control (ovladanie)
client.subscribe("Settings") #odoberanie sprav s temou Settings (nastavenia)
client.on_message = on_message
time.sleep(1)
client.loop_forever()
```

Výpis 4.1: Ukážka časti zdrojového kódu pre implementáciu MQTT klienta

4.2.5 JavaScript

Je programovací jazyk ktorý pridáva webovým aplikáciam interáktívne správanie. Tento objektovo orientovaný jazyk pracuje na strane klienta v prehliadači. Užívateľ pošle požiadavok na server, a server späť zašle HTML a skript, ktorý je následne spracovaný prehliadačom. V projekte je vyuzívaný na ovládanie tlačidiel na webe, a spolu s ajaxom tvoria dynamické načítavanie dat.

4.3 Hlavný program

Po spustení programu sa automaticky zapne webový flask server na ktorý sa môže uživateľ pripojiť. Na hlavnej stránke sú zobrazené namerané teploty a stav komponentov. Uživateľ má možnosť pomoocu webového rozhrania ručne kontrolovať podavač, čerpadlo a ventilátor. V prípade zapnutia kotla do pracujúceho stavu sa v programe spustia vlákna, ktoré ovládajú celý proces regulácie. Riešenie pomocou vlákien bolo zvolené z dôvodu toho, že celý program beží asynchrónne. Synchronizácia týchto vlákien je zabezpečená porovnávaním stavov z konfiguračného súboru. Napríklad vlákno ovládajúce proces spaľovania sa ukončí v prípade že sa odstaví kotol.

Pri ukončení programu sa spustí funkcia ktorá odstaví kotol a vypne všetky elektronické súčiastky. Cieľom je zabezpečenie súčiastok v prípade chybavého stavu.

```
# vypnutie vsetkych komponentov
def exit_handler():
    fm.setBoilerPumpState('OFF')
    fm.setValveState('CLOSE')
    fm.setBoilerStatus('OFF')
    status.stop()
atexit.register(exit_handler)
```

Výpis 4.2: Ukážka zdrojového kódu pre handler pri ukončení aplikácie

Keďže zapnutie čerpadla je asynchrónna operácia, funkcia sa spúšťa na vlákne. Pracuje vo vlastnom cykle a kontroluje sa teplota zapnutia čerpadla a aktuálna teplota kotla, poprípade stav. Implementovaná je taktiež hysterézia, takzvaná odchýlka ktorá sa odpočítava od teploty zapnutia čerpadla. Pred spustením funkcie sa nastavý stav na ON a po ukončení na OFF.

```
def boilerheatpump():
    startBoilerHeatPump()
    fm.setBoilerPumpState('ON')
    timestamp = time.time()
    while True:
        temp = temperatures.getBoilerTemp()
        print(temp, fm.getPumpTemp(), fm.getPumpDifferenceTime())
        if temp <= fm.getPumpTemp() - fm.getPumpDifferenceTime() or time.time() -
            timestamp > fm.getPumpDelay():
            if temp <= fm.getPumpTemp() - fm.getPumpDifferenceTime():
                break
    stopBoilerHeatPump()
    fm.setBoilerPumpState('OFF')
```

Výpis 4.3: Ukážka zdrojového kódu pre zapnutie čerpadla

Práca so súborom conf.json. Funkcia funguje štýlom načítania konfiguračného súboru, zmenenie danej hodnoty, a prepísanie súboru s novou hodnotou. V prípade zapnutého debugu sa zmena stavu zapíše do logeru

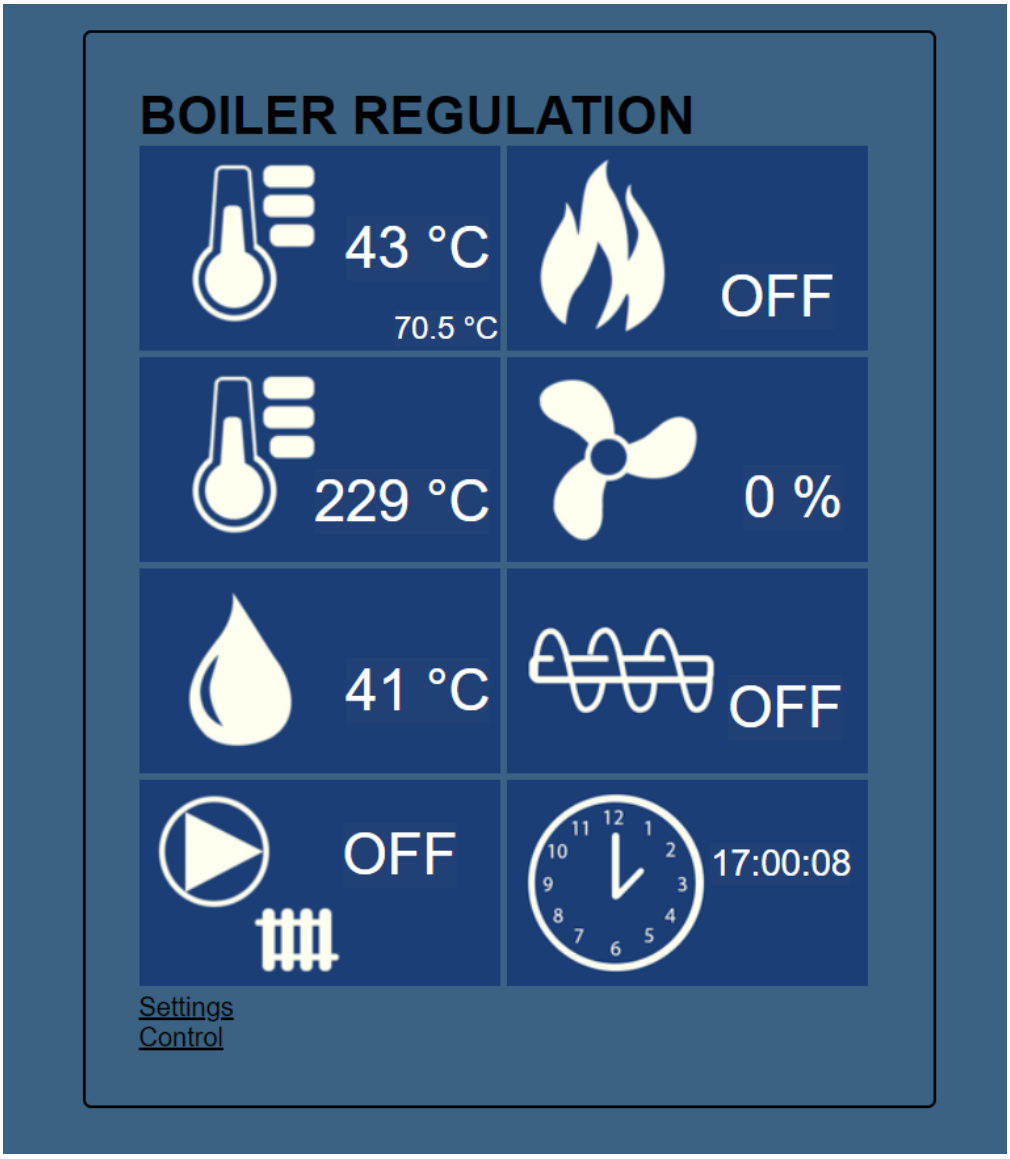
```
def setBoilerPumpState(state):
    try:
        a_file = open("conf.json", "r")
        json_object = json.load(a_file)
        a_file.close()
        json_object['settings']['PROGRAM']['PUMP_STATE'] = state
        a_file = open("conf.json", "w")
        json.dump(json_object, a_file)
        a_file.close()
        if isDebugOn(): logging.info('Cerpadlo kotla nastavene na stav: ' + state)
    except OSError:
        print('Cannot open conf.json')
```

Výpis 4.4: Ukážka zdrojového kódu pre nastavenie stavu čerpadla

.

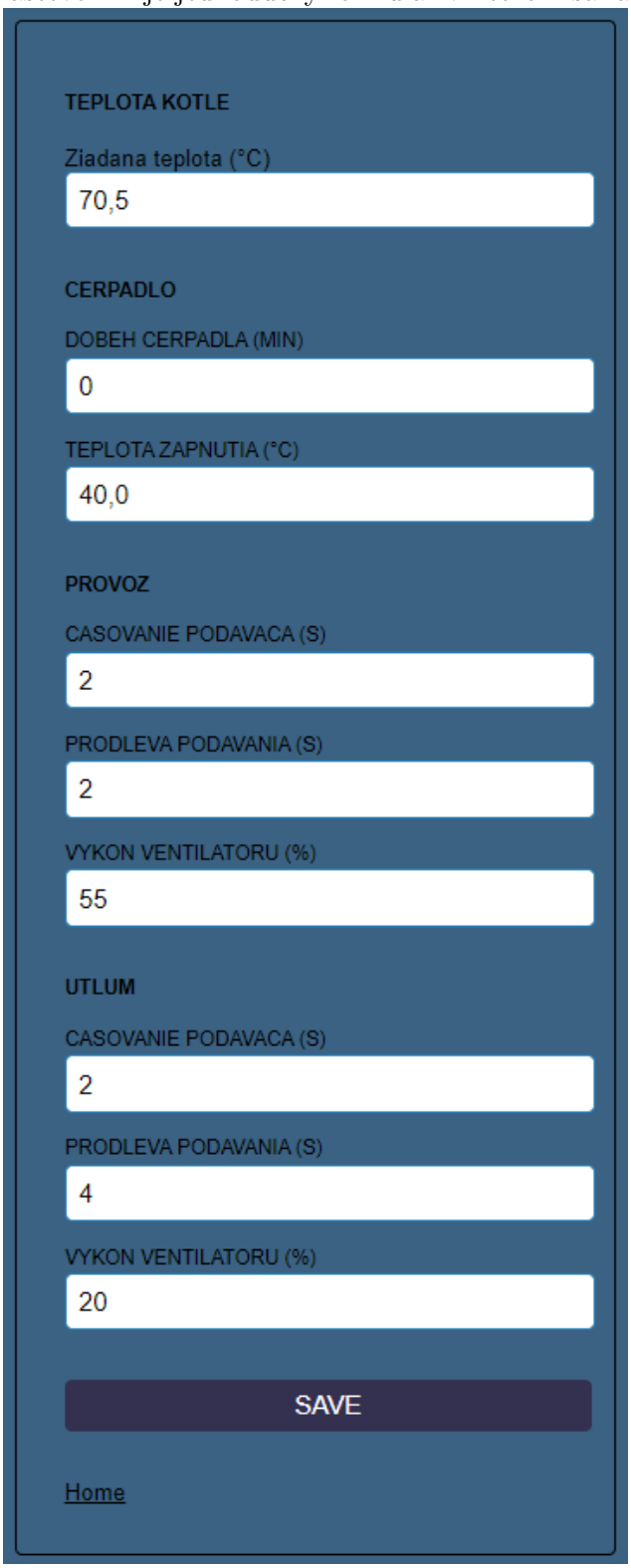
4.4 Webové rozhranie

Na domovskej stránke sú vidno aktuálne hodnoty a stavy použitých zariadení. Dáta sú načítane dynamicky bez nutnosti opakovaného načítania stránky.



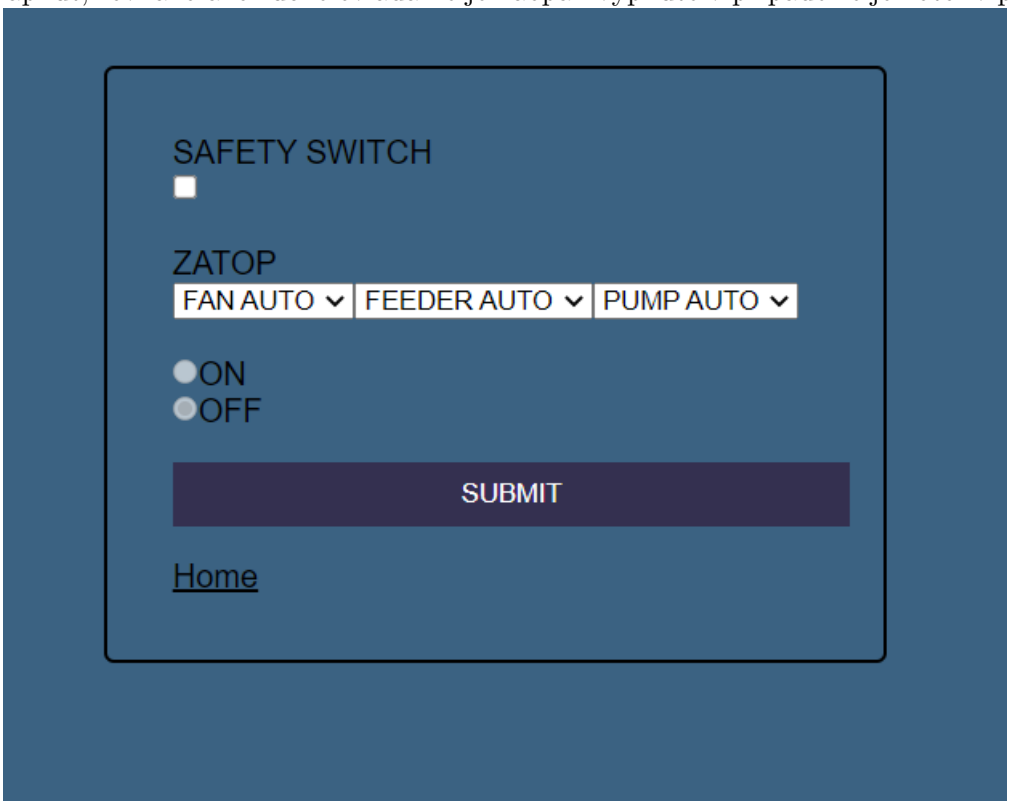
Obr. 4.3: Hlavná stránka

Na stránke s nastevením je jednoduchý formulár v ktorom sa ladí daný kotol.



Obr. 4.4: Stránka nastavení

Na stránke s ovládaním je možné vidieť kontrolný panel. Po odkliknutí bezpečnostného prepínača sa uživateľovi sprístupní možnosť zapnutia kotla do prevádzky. V sekcii zátop je možné koponenty ovládať ručne. Stavy sú ošetrené, ako napríklad v prípade že program riadenia kotla beží, už sa nedá zas zapnúť, rovnako ako ručné ovládanie je naopak vypnuté v prípade že je kotol v prevádzke.

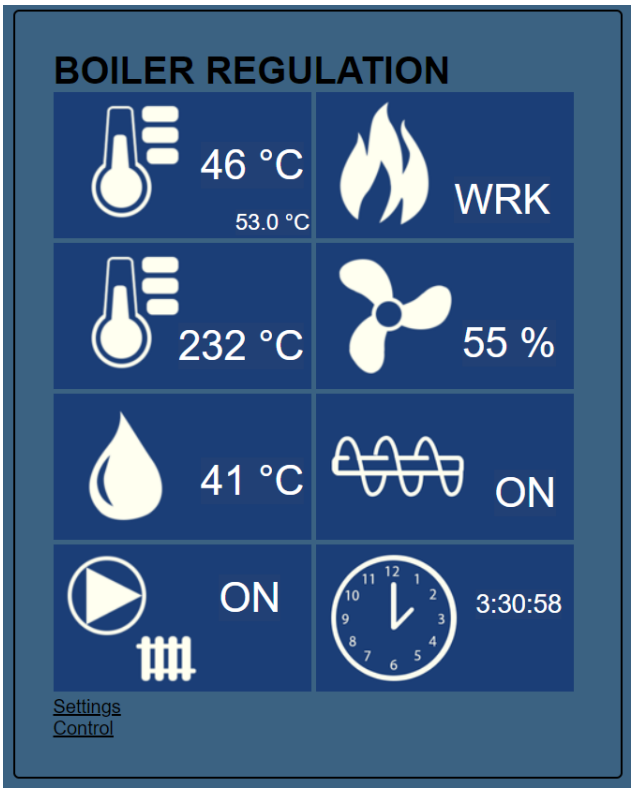


Obr. 4.5: Stránka ovládania

Kapitola 5

Výsledné testovanie riadiacej jednotky

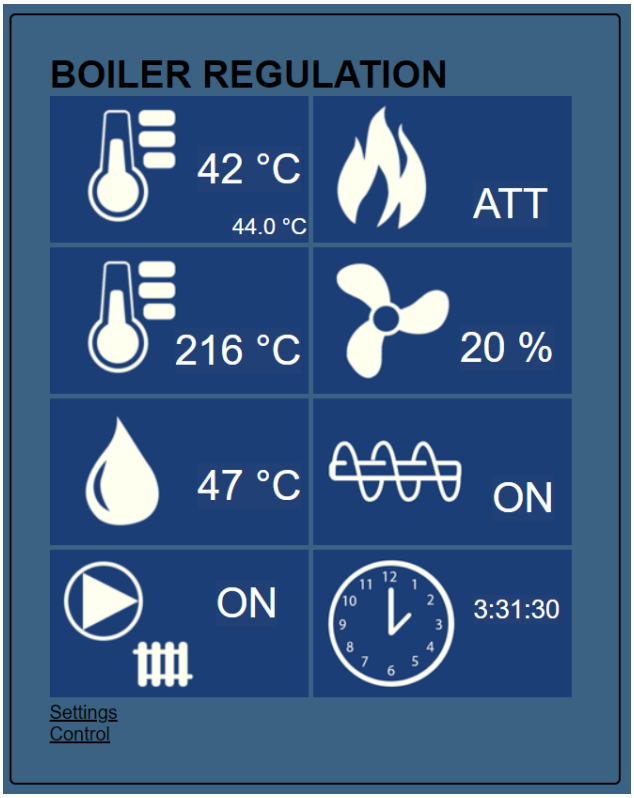
Nasledujúca kapitola sa venuje testovaniu diela riadiacej jednotky pre automatické kotle. Z dôvodu pandémie covid 19, a lockdownu v Českej a Slovenskej republike nebolo možné jednotku odtestovať v plnom nasadení pri reálnom kotli. Nasledujúce testy sú spravené len simulovane.



Obr. 5.1: Riadiaca jednotka v režime provoz

Po zapnutí režimu provoz, sa nastaví ventilátor na daný výkon a spolu s podavačom pracuje v nastavenom cykle. Môžne vidieť že status (ikonka s ohňom) je nastavený ako WRK- Working.

Úloha tohoto režimu je dosiahnuť žiadanú teplotu kotla (teplota kotla podľa obrázka je 46°C a je potrebná 53°C). V tomto prípade sa zaplo taktiež čerpadlo (viz obrázok 4.3 teplota zapnutia v sekcii čerpadlo) okruhu a klesne až ked teplota kotla padne pod nastavených 40°C.



Obr. 5.2: Riadiaca jednotka v režime útlum

Kotol prejde do režimu útlum (ATT - ATTENUATION) v prípade že žiadaná teplota (44°C) klesne, a je menšia než teplota kotla. Zníži sa výkon ventilátora a podávacie cykly sú viac zriedkavé. V prípade veľkého poklesu teploty sa zas zapne režim provoz. Takto funguje celý princíp regulácie na pevnú teplotu v kotli. Čerpadlo stále beži nezávisle od režimu- zastaví sa keď teplota kotla klesne pod 40°C alebo ho uživateľ vypne ručne, poprípade sa odstavý kotol pri chybovom stave.

Ukážka logovania pri minimálnych oneskoreniach. Prechod kotla zo stavu útlum do stavu provoz a vypnutie kotla.

```
2021-04-30 11:58:44 INFO-Stav utlum-> teplota kotla: 44 C, Ziadana teplota: 45 C
2021-04-30 11:58:44 INFO-Podavac nastaveny na stav: ON
2021-04-30 11:58:46 INFO-Podavac nastaveny na stav: OFF
2021-04-30 11:58:48 INFO-Boiler nastaveny na stav: ATTENUATION
2021-04-30 11:58:48 INFO-Nastavenie vykonu vetraka na 20 %
2021-04-30 11:58:48 INFO-Boiler nastaveny na stav: WORKING
2021-04-30 11:58:48 INFO-Nastavenie vykonu vetraka na 55 %
2021-04-30 11:58:48 INFO-Stav provoz-> teplota kotla: 40 C, Ziadana teplota: 45 C
2021-04-30 11:58:48 INFO-Podavac nastaveny na stav: ON
2021-04-30 11:58:50 INFO-Podavac nastaveny na stav: OFF
2021-04-30 11:58:51 INFO-Boiler nastaveny na stav: OFF
2021-04-30 11:58:51 INFO-Boiler nastaveny na stav: OFF
2021-04-30 11:58:51 INFO-Podavac nastaveny na stav: OFF
2021-04-30 11:58:51 INFO-Nastavenie vykonu vetraka na 0 %
2021-04-30 11:58:51 INFO-Odstavenie kotla
```

Obr. 5.3: Ukážka logovacích hlášok pri prevádzke kotla

Kapitola 6

Záver

Cieľom mojej práce bolo zoznámenie sa s platformou Raspberry pi a zanalyzovať postup riešenia. V analýze som popísal použitú platformu a zdôvodnenie výberu komponentov. Podľa tejto analýzy bol ďalší krok navrhnúť hardwarové a softwarové riešenie. Základným pilierom programu bol zvolený programovací jazyk Python. Aplikácia má možnosť distribuovaného nasadenia, to znamená že dokáže ovládať viacero kotlov naraz. Ovládanie a monitoring sa zobrazuje na webovom rozhraní na ktoré sa môže uživateľ jednoducho pripojiť. Výsledný produkt je open source riadiaca jednotka pre automatické kotle, pripravená na ovládanie a kontrolovanie kotla na tuhé palivo.

Literatura

1. *Kotol na kvapalné palivo* [online] [cit. 2021-04-30]. Dostupné z : https://engineer.decorexpro.com/sk/otoplenie/kotly/kotly-otopleniya-na-zhidkom-toplive.html.
2. *Kotol na plynné palivo* [online] [cit. 2021-04-30]. Dostupné z : https://maxibyvanie.sk/funguje-plynovy-kotol/.
3. *Kotol na tuhé palivo* [online] [cit. 2021-04-30]. Dostupné z : http://www.techpark.sk/technika-11-2009/automaticke-kotle-na-spalovanie-tuhych-paliv-realna-moznost-lacnejsieho-a-komfortneho-vykurovania.html.
4. *Raspberry pi* [online] [cit. 2021-04-30]. Dostupné z : UPTON,%20Eben%20a%20Gareth%20HALFACREE.%20Raspberry%20Pi:%20u%C5%BEivatelsk%C3%A1%20p%C5%99%C3%ADru%C4%8Dka.%20Brno:%20Computer%20Press,%202013.%20ISBN%20978-80-251-4116-8..
5. *Gpio piny* [online] [cit. 2021-04-30]. Dostupné z : https://www.raspberrypi.org/documentation/usage/gpio/.
6. *Senzory* [online] [cit. 2021-04-30]. Dostupné z : https://www.abebooks.com/9781784393618/Raspberry-Pi-Sensors-Gajjar-Rushi-1784393614/plp.
7. *Relé* [online] [cit. 2021-04-30]. Dostupné z : https://tutorials-raspberrypi.com/raspberry-pi-control-relay-switch-via-gpio/.
8. *Teplomery dallas* [online] [cit. 2021-04-30]. Dostupné z : https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf.
9. *Hallov senzor* [online] [cit. 2021-04-30]. Dostupné z : http://www.epitran.it/ebayDrive/datasheet/20.pdf.
10. *Prevodník max6675* [online] [cit. 2021-04-30]. Dostupné z : https://datasheets.maximintegrated.com/en/ds/MAX6675.pdf.
11. *PWM* [online] [cit. 2021-04-30]. Dostupné z : https://www.eleinmec.com/article.asp?28.
12. *Python* [online] [cit. 2021-04-30]. Dostupné z : https://www.amazon.com/Learning-Python-Raspberry-Alex-Bradbury/dp/1118717058.

13. *Framework flask* [online] [cit. 2021-04-30]. Dostupné z : https://web.archive.org/web/20180517082208/http://flask.pocoo.org/extensions/.

14. *Ajax* [online] [cit. 2021-04-30]. Dostupné z : https://www.pluralsight.com/guides/work-with-ajax-django.

15. *Protokol mqtt* [online] [cit. 2021-04-30]. Dostupné z : https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/.